

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Application Serial No.: 10/849,294  
Applicants: Plante  
Filing Date: May 19, 2004  
Art Unit: 2192  
Examiner: Wei, Zheng  
Confirmation No.: 5608  
Title: Method for Synchronization of Concurrently Modified  
Interdependent Semi-Derived Artifacts

Commissioner for Patents  
Mail Stop Appeal Brief - Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

In response to the non-final Office Action, dated January 7, 2010, rejecting pending claims 1-4, and in support of the Notice of Appeal received by the U.S. Patent and Trademark Office on April 6, 2010, Appellant hereby submits this Appeal Brief to the Board of Patent Appeals and Interferences. The previously paid appeal brief fee is being applied to this new Appeal Brief in accord with Item 1, section (2) on page 2 of the subject Office Action. Appellant believes that no fee is due; however, should any fee be required by the filing of this Appeal Brief, authorization is herein granted to apply such fee and any other fees or credits due in this case, to Deposit Account No. 122158. Appellant respectfully requests reconsideration and reversal of the Examiner's rejections of the pending claims.

## TABLE OF CONTENTS

|   |       |
|---|-------|
| Real Party in Interest .....                        | 3     |
| Related Appeals and Interferences .....             | 3     |
| Status of Claims .....                              | 3     |
| Status of Amendments .....                          | 4     |
| Summary of Claimed Subject Matter .....             | 4-5   |
| Grounds of Rejection to be Reviewed on Appeal ..... | 5     |
| Argument .....                                      | 6-11  |
| Claims Appendix .....                               | 12-13 |
| Evidence Appendix .....                             | 14    |
| Related Proceedings Appendix .....                  | 14    |

REAL PARTY IN INTEREST

The Real Party in Interest is International Business Machines Corporation, the owner of all rights of this patent application by virtue of an assignment recorded at reel and frame number 014994/0182.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

The patent application as originally filed included claims 1-18. Claims 1, 7 and 15 were amended and claims 5, 6, 9-14, 17 and 18 were canceled in an Amendment and Response dated June 13, 2007 in response to a non-final Office Action mailed April 17, 2007. Claims 3 and 4 were amended in an Amendment and Response dated November 16, 2007 in response to a non-final Office Action mailed August 22, 2007. Claim 1 was amended and claims 7, 8, 15 and 16 were canceled in an Amendment and Response dated April 14, 2008 in response to a final Office Action mailed February 7, 2008. A Request for Reconsideration in which no claim amendments were made was filed October 6, 2008 in response to a non-final Office Action mailed July 17, 2008. A Request for Reconsideration in which no claim amendments were made was filed March 4, 2009 in response to a non-final Office Action mailed January 12, 2009. The final Office Action mailed June 22, 2009 rejects claims 1-4. Claims 1-4 were the subject of an appeal filed on October 1, 2009. Subsequently, prosecution was reopened in a non-final Office Action mailed January 1, 2010. The non-final Office Action rejects claims 1-4. Claims 1-4 remain pending in the application and are the subject of this appeal.

### STATUS OF AMENDMENTS

No amendments or other forms of response have been filed subsequent to the mailing of the final Office Action mailed January 7, 2010.

### SUMMARY OF CLAIMED SUBJECT MATTER

#### Independent claim 1

Appellant's invention, as recited in claim 1, features a method for synchronizing a first artifact and a second artifact. (For example, the first and second artifacts can be a software model UML file and a third-generation language (3GL) code file. See page 8, line 11 to page 9, line 5.) The first and second artifacts are interdependent. See page 9, lines 2-5. Each artifact 14 and 18 is modified independent of a modification to the other artifact after a last synchronization. See the independently-generated versions of the artifacts in the timeline shown in FIG. 1 subsequent to the single depicted synchronization (SYNC) with descriptive text at page 9, lines 6-16. Also see the independently-generated versions of the artifacts in the timelines shown in FIG. 2 and FIG. 3 subsequent to the first synchronization but before a second synchronization. Each of the first and second artifacts has a plurality of elements and is of a different format. See page 8, lines 11 and 12 (artifact is a file); page 10, lines 1 and 2 (artifacts have elements); and page 8, line 12 to page 13, line 5 (UML and 3GL artifacts or other related artifacts that require application of a forward engineering operation or a reverse engineering operation to transform from one format to the other format).

A reverse engineering operation is performed to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact. See step 110 in FIG. 3 and page 10, line 15 to page 11, line 2. Also see FIG. 5 and page 11, lines 3-14 describing how the particular limitations recited for performing the claimed reverse engineering operation are different from a conventional reverse engineering operation. The temporary artifact and a latest version of the first artifact are merged to create a synchronized version of the first artifact. See step 140 in FIG. 3 and page 11, lines 15-19. A forward engineering operation is performed to generate a synchronized version of the second artifact that has all the elements of the latest version of the second artifact and that also has all the elements of the synchronized version of the first artifact transformed as the second artifact. See step 150 in FIG. 3 and page 12, lines 3-6. Also see FIG. 6 and page 12, lines 10-20 describing how the particular limitations recited for performing the claimed forward engineering operation are different from a conventional forward engineering operation.

#### GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The non-final Office Action mailed January 7, 2010 issued the following rejections:

- I. Claims 1, 2 and 4 are rejected under 35 U.S.C. § 103(a) as being anticipated by U.S. Patent No. 6,502,239 B2.
- II. Claim 3 is rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2 in view of U.S. Patent No. 6,038,393.

The grounds of rejection to be reviewed on appeal are grounds I and II as applied to claims 1-4.

## ARGUMENT

### Rejection under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2

The non-final Office Action rejects claims 1, 2 and 4 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2 to Zgarba et al. (hereinafter “Zgarba”).

By way of a general background, Zgarba discloses a method of round-trip engineering source code from a software model. The disclosure includes the application of forward engineering to previously reverse engineered code in the software model so that the updated source code generated by the forward engineering does not include any changes to the code that were not made in the software model. (See, e.g., Abstract.) “Importantly, existing source code not affected by changes in the software model is left unchanged.” (Col. 7, lines 4-6)

### Claim 1

Zgarba discloses round-trip software engineering in which a software application is reverse engineered into a software model, the software model is changed, and then the code represented by the software model is re-coded. (Col. 1, lines 5-10) Zgarba addresses how to perform the round-trip engineering that allows the software model to be kept synchronized with the source code (or equivalent objects) without the use of code markers which would otherwise clutter the software code. (Col. 2, lines 9-12) Zgarba stresses the value of “proper round-trip engineering” of a software project:

Importantly, when a software project is reverse engineered from source code, and the software model generated is forward engineered, the resultant code should be essentially the same, regardless of what information in the source code is represented in the software model. This allows for proper “round-trip” engineering of the software project, so that the software model and the source code can be kept synchronized, updates in both the source code and the software model will be maintained when generating the source code from the software model and vice versa, and comments and other features of the source code not represented in the software model do not get moved around or even disappear in the round-trip process. (Col. 4, lines 18-30)

Appellant's claim 1 relates to a method for synchronizing a first artifact and a second artifact. Claim 1 recites, in part, "each artifact being modified independent of a modification to the other artifact after a last synchronization."

The subject Office Action states on page 6 that Zgarba discloses Appellant's "method for synchronizing a first artifact (source code) and a second artifact (software model), the first and second artifacts being interdependent and one of the artifacts being modified independent of a modification to the other artifact after a last synchronization." The Office Action refers by way of example to Fig. 4 and related text to support this statement. Appellant respectfully disagrees with the statement because the limitation is not disclosed in the figure and its related text, or elsewhere in Zgarba. FIG. 4 and the related text simply relate to a round-trip engineering process that is used for synchronization in a typical sense in which (1) a change is made to the software model and the software code is then synchronized to the changed software model or (2) a change is made to the software code and the software model is then synchronized to the changed software code. However, Zgarba does not address, either in the cited figure and text, or elsewhere in the disclosure, a situation in which both artifacts (i.e., the software model and the software code) are changed independently between synchronizations (i.e., "concurrent modified") and how to subsequently synchronize the changed software model and the changed software code. Thus Zgarba does not teach or suggest "each artifact being modified independent of a modification to the other artifact after a last synchronization."

In contrast to the above, on page 3 of the Office Action, the Examiner agrees that "Zgarba discloses a method to synchronize "source code" with the modified "software model", but does not explicitly disclose synchronizing two "source code" and "software model" that are both modified." The Examiner continues by taking the position that the "present application as disclosed are merely a three steps synchronization process ..." and that "[t]he key point is that there is no much difference to merge the artifacts that can be modified or not." Appellant respectfully disagrees. Application of standard synchronization steps cannot achieve the desired synchronization of two artifacts that are independently modified between synchronizations and the step of merging, in isolation, does not have any bearing on the ability to synchronize independently modified artifacts. Moreover, Zgarba provides no specific teaching and no

suggestion as to how one would modify any of the disclosed steps to be able to handle a situation in which both artifacts are changed after the last time the artifacts were synchronized.

Notwithstanding the above arguments, Appellant's claim 1 further recites "performing a reverse engineering operation to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact." (Emphasis added) Thus the temporary artifact has elements that are in the format of the first artifact.

To show this limitation (except that the Examiner has exchanged the phrases "reverse engineering" and "forward engineering" in the limitation) the Office Action refers on page 7 to Figs. 9A, 9B and 9C of Zgarba, and the related text. Appellant notes that Zgarba explains that meta-models can be "generated from the modified software model shown in FIG. 8 and the original source code shown in FIG. 5 are shown in FIGS. 9A and 9B, respectively. The meta-models might for example be generated in CDIF format ...." (Col. 7, lines 32-38; also see Col. 5, lines 1-17 regarding the CASE Data Interchange Format (CDIF)) Zgarba specifically states that "[m]appings from each specific environment to the generic meta-model are developed to ensure that no data is lost and the appropriate information is compared." (Col. 7, lines 42-45) Thus Zgarba explicitly indicates that the meta-model is neither in the form of source code format nor in the model format. Instead, the meta-model is in an intermediate format that accepts data from the source code and the model, that is, an intermediate format that accepts data from the first artifact and the second artifact. Appellant's recited method avoids any intermediate format.

On page 4 of the Office Action, the Examiner states the position that "Zgarba's meta-model as illustrated in Fig. 9C contains all the elements in code version and model version which are both in the model format (meta-model)." Appellant respectfully disagrees with the Examiner's conclusion that the updated/generated meta-model in Fig. 9C is the same as Appellant's recited temporary artifact. The Examiner's position statement is consistent with Appellant's argument above, that is, according to Zgarba, the elements of the code and model versions exist in the format of the meta-model which is not the same as the code format or the model format. Thus Zgarba does not teach or suggest "a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact."



Appellant addresses here the last paragraph on page 5 of the subject Office Action. Specifically, the Examiner states:

At page 8, third paragraph, the Appellant submits that no teaching or suggestion is presented to show the reverse engineering being applied to artifacts that are independently modified after a last synchronization. However, it should be noted that reverse engineering as recited is merely for the purpose of merging two different artifacts that both are required in the same format. Zgarba discloses a version of software code and a version of software model are converted (reverse or forward engineering) to meta-models (fig.9A and fig.9B) wherein at least one artifact (software model) is independently modified by adding "doThat(): void" (see for example, Fig.9B). Therefore, as Zgarba disclosed by using reverse engineering to convert both software code and model to meta-models, it is obvious that such reverse engineering can be applied to any artifacts including independently modified to generate the meta-model for the purpose of merging or synchronization as suggested by Zgarba.

Appellant respectfully disagrees. The Examiner's conclusion is based on the disclosure of a reverse engineering operation to convert to a meta-model format without any indication as to why that process makes it applicable to independently modified artifacts. Appellant again emphasizes that simply using reverse engineering to convert both the software code and the model is insufficient when both artifacts are changed after the last synchronization. Appellant notes that nowhere in Zgarba is it taught or suggested that the updates in the source code and the software model are made independently between synchronizations.

In sum, Appellant notes that Zgarba addresses a round-trip process to synchronize a software model and code so that changes made to one are reflected in the other. In contrast, Appellant's invention as recited in method claim 1 is based on processing artifacts that are both modified, independent from modifications made to the other, between synchronizations. For the reasons set forth above, Appellant respectfully submits that Zgarba does not teach or suggest all the limitations recited in claim 1 and therefore that claim 1 is allowable over Zgarba.

#### Claims 2 and 4

Claims 2 and 4 depend from allowable claim 1 and incorporate all of the limitations of claim 1. Thus claims 2 and 4 are allowable over Zgarba for at least the reasons provided above with respect to claim 1.

#### Rejection under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,502,239 B2 in view of U.S. Patent No. 6,038,393

The non-final Office Action rejects claim 3 under 35 U.S.C. § 103(a) as being unpatentable over Zgarba in view of U.S. Patent No. 6,038,393 to Iyengar et al. (hereinafter "Iyengar").

#### Claim 3

The final Office Action relies on Iyengar to show a limitation introduced in dependent claim 3 that is not taught or suggested by Zgarba. Claim 3 depends from allowable claim 1 and incorporates all of the limitations of claim 1. The disclosure of Iyengar does not cure the deficiencies noted above with respect to the limitations in claim 1 that are not taught or suggested by Zgarba. Accordingly, claim 3 recites limitations that are not taught or suggested by the cited references, whether taken alone or in proper combination. Thus Appellant respectfully submits that claim 3 is patentable over Zgarba and Iyengar.

In view of the arguments made herein, Appellant submits that the application is in condition for allowance.

Respectfully submitted,

Date: May 26, 2010  
Reg. No. 41,047

Tel. No.: (508) 303-2003  
Fax No.: (508) 303-0005

/William G. Guerin/  
William G. Guerin  
Attorney for Appellant  
Guerin & Rodriguez, LLP  
5 Mount Royal Avenue  
Marlborough, MA 01752

CLAIMS APPENDIX

1. A method for synchronizing a first artifact and a second artifact, the first and second artifacts being interdependent and each artifact being modified independent of a modification to the other artifact after a last synchronization, the first and second artifacts each having a plurality of elements and being of different formats, the method comprising:

performing a reverse engineering operation to generate a temporary artifact having all the elements of a last synchronized version of the first artifact and having all the elements of a latest version of the second artifact transformed as the first artifact;

merging the temporary artifact and a latest version of the first artifact to create a synchronized version of the first artifact; and

performing a forward engineering operation to generate a synchronized version of the second artifact having all the elements of the latest version of the second artifact and having all the elements of the synchronized version of the first artifact transformed as the second artifact.

2. The method of claim 1 wherein one of the first and second artifacts is a software model artifact and the other of the first and second artifacts is a code artifact.

3. The method of claim 2 wherein the software model artifact is a Unified Modeling Language file.

4. The method of claim 2 wherein the code artifact is a third-generation language source file.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.